Scenario 1: Logging

How would you store your log entries?

AW: I would use cookie to store log entries

How would you allow users to submit log entries?

AW: I would like to allow users to submit log entries via a form

How would you allow them to query log entries?

AW: Just query the databased where logs will be stored.

How would you allow them to see their log entries?

AW: Add a function to print the log information.

What would be your web server?

AW: Node.js or Apache

Scenario 2: Expense Reports

How would you store your expenses?
AW: I would like to store them in mongoDB.

What web server would you choose, and why?

AW: Node.js, because it’s a full-stack JavaScript for serving both client side and the server side application and it is easy to use.

How would you handle the emails?

AW: “npm install nodemailer”, install nodemailer, Nodemailer is a single module with zero dependencies for Node. js, designed for sending emails.

How would you handle the PDF generation?

AW: I would like to use LaTex to handle the PDF generation.

How are you going to handle all the templating for the web application?

AW: I would like to use react to handle the templating.

Scenario 3: A Twitter Streaming Safety Service

Which Twitter API do you use?

AW: Twitter API v2. URL: https://developer.twitter.com/en/docs/twitter-api/tutorials

How would you build this so its expandable to beyond your local precinct?

AW: Using the filtered stream and sampled stream endpoints, I can listen for important events from public Tweets as those events unfold in real-time. The filtered stream endpoint returns a specific set of Tweets matching the criteria that you set while the sampled stream endpoint returns about a 1% random sample of all public Tweets in real-time.

What would you do to make sure that this system is constantly stable?

AW: I would like to add some error handling processes to throw errors, and make sure the function can cover most of the events I want to listen.

What would be your web server technology?

AW: Apache.

What databases would you use for triggers?

AW: MongoDB.

For the historical log of tweets? How would you handle the real time, streaming incident report?

AW: I can get the data from the api every 30 seconds to keep the event fresh.

How would you handle storing all the media that you have to store as well?

AW: I would like to storing them in mongoDB and query them later.

What web server technology would you use?

AW: Apache.

Scenario 4: A Mildly Interesting Mobile Application

How would you handle the geospatial nature of your data?

AW: I will first get the location of users then store the images data

How would you store images, both for long term, cheap storage and for short term, fast retrieval?

AW: For the long term storage I will use the mongoDB, for cheap and short term storage, I will use a server.

What would you write your API in?

AW: Login function and image process function.

What would be your database?

AW: MongoDB